

N87 - 22610

DIGITAL AUTONOMOUS TERMINAL
ACCESS COMMUNICATION SYSTEM
(DATAC)

Stanley M. Novacki, III
Ohio University
Athens, Ohio

In order to accommodate the increasing number of computerized subsystems aboard today's more fuel-efficient aircraft, the Boeing Company has developed the DATAC (Digital Autonomous Terminal Access Control) bus to minimize the need for point-to-point wiring to interconnect these various systems, thereby reducing total aircraft weight and maintaining an economical flight configuration. The DATAC bus is essentially a local area network providing interconnections for any of the flight management and control systems aboard the aircraft. The bus itself is passive: a twisted-pair, coax, or fiber-optic line with stub connectors for each DATAC terminal, the device which interfaces the various subsystems with the bus. DATAC uses a line-topology and CSMA (Carrier Sense, Multiple Access) protocol, much like the commercial Ethernet local-area network proposed by Intel and Xerox, which is in use in many large offices today. The main purpose of the Bus Monitor Unit is to receive all data which appear on the bus, reconstitute it into "real-word" parameters (e.g., altitude, heading, etc.), and display it in an easily interpreted form. This provides an indication of the overall operation of the bus without the liability of a total system failure caused by a central controller-type device, which could be disastrous at 35,000 feet.

The task of developing a Bus Monitor Unit can be broken down into four subtasks: 1) Providing a hardware interface between the DATAC bus and the Z8000-based microcomputer system to be used as the Bus Monitor; 2) Establishing a communications link between the Z8000 system and a CP/M-based computer system, which will be used as the Bus Monitor Unit console; 3) Generation of data reduction and display software to output data to the console device; and 4) Development of a DATAC Terminal Simulator to facilitate testing of the hardware and software which transfer data between the DATAC bus and the operator's console in a near real-time environment.

A. DATAC Terminal Interface Card:

The subsystem interface used on the DATAC bus terminals used the ZILOG Zbus system interconnect specification. However, the Bus Monitor test system uses the IEEE-696 bus. An interface card has been developed that will translate the two sets of signals, interchangeably, to allow each system to transfer data to or from the other in the same manner that a DATAC terminal will communicate with a conventional bus subsystem.

B. Software Communications Driver for Z8000 to VT-180:

A program will be written in Z8000 assembly language to detect new data placed in the Bus Monitor memory by the DATAC terminal, reduce it to a format which allows for some kind of error detection (for example, hex-to-ASCII conversion with parity) and transmit it via an RS-232C line to the Digital Equipment Corporation VT-180 microcomputer serving as the BusMon system console.

C. DATAC Bus Data Analysis and Display:

The actual reduction of data from the DATAC bus initially will be performed by a VT-180 using the "C" language. There are several reasons for

this approach: the Z8000 microcomputer will eventually perform most of the operations planned for the Bus Monitor Unit. At this time, however, few development tools can be used on this configuration of a Z8000 system. Further, the Bus Monitor processor system does not currently support any type of mass storage devices, printers, or other peripherals useful for prototype work. The intent is, then, to develop the Bus Monitor software using the standard "C" language under the CP/M operating system for which there are a variety of software development tools. Once the software is completed and tested, any functions that can be supported by the Z8000 system hardware can be transported to that system via a "C" cross-assembler targeted for that system. Also, the ease of writing involved algorithms in a higher-level language facilitates the programming of data analysis tasks, ordinarily cumbersome to write in assembly language. Additionally, a higher level language is portable; the Bus Monitor functions can be transported to a system based on another processor with little additional programmer time should Boeing and NASA decide that a different hardware configuration is needed. Thus, the investment in software is guarded against obsolescence in the event of a system hardware change or eventual upgrade.

D. DATAAC Terminal Simulation:

A DATAAC terminal simulator is needed because, at present, the only operational model of the DATAAC subsystem interface exists on an active DATAAC system. A programmable terminal simulator offers the flexibility needed to test and properly debug the Bus Monitor Unit hardware and software under a variety of conditions, something not easily done using an actual DATAAC system.

CURRENTLY ENVISIONED BUS MONITOR FUNCTIONS

In order for the BusMon unit to provide a useable picture of the operation of the DATAAC bus, it must be able to evaluate the data which appears on the bus and present it in an operator recognizable format. Since bus protocol violations and communications errors (e.g., parity errors, missing words, etc.) are detected by the DATAAC terminal hardware and are never seen by any bus subsystem, the Bus Monitor software will address itself to problems determined by the data content of messages from other DATAAC terminals:

1. Echo check of Terminal/System Interfaces: BusMon will verify proper operation of other subsystem hardware by means of a transmit block after receive block directive from the BusMon to the system in question. This assures integrity of RAM buffers and other terminal interface devices.
2. Major/Minor Frame Usage: The BusMon will be equipped with the bus system transmission schedule. If a system fails to transmit in its allotted time frame(s), the monitor can signal a device failure.
3. Data Stream Interpretation: The format of the data stream allows the BusMon to identify the source and information content

of the serial message. A useful feature of this function is to provide a check on the reasonability of data being placed on the bus by the various other bus subsystems. Reconstitution of bus data by an independent system provides another check on the accuracy of other devices operating on the bus. For example, if a display unit shows unreasonable data items, the problem could lie in the data source or in the display unit. By monitoring the data intended for the display, BusMon can help locate potential hardware problems.

4. **Bus Status:** The most passive mode of operation simply displays hexadecimal data extracted from the bus, formats and displays it on the BusMon console to provide transmitter ID, word count, and other raw data items.

WHAT IS DATAC?

ORIGINAL PAGE IS
OF POOR QUALITY

- Serial data bus for use aboard aircraft
- Various A/C systems communicate via a single twisted-pair, coaxial, or fiber optic line
- No central controller (As used by MIL STD 1553B, system relatively free from central failures)
- Uses Ethernet-like CSMA/CD access protocol
- Line topology of bus allows for easy addition/deletion of systems for maintenance or expansion

SUMMARY OF DATAC ADVANTAGES

*WEIGHT REDUCTION- a single bus could serve a minimum-configuration system

*REDUCED INSTALLATION COSTS AND COMPLEXITY- 1-5 bus lines as opposed to one bus line per system inter-connect

*IMPROVED DIAGNOSTICS AND MAINTENANCE- subsystems could be programmed to provide status information to the bus monitor

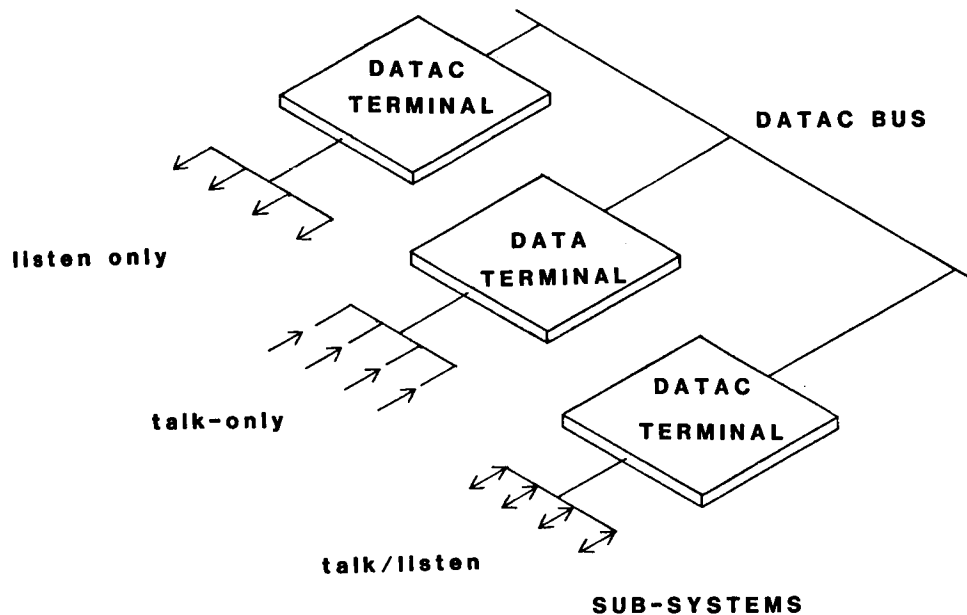
*SIMPLE TO MODIFY- line topology of bus allows for easy addition and removal of subsystems

*POWERFUL LOCAL PROCESSING CAPABILITY- a subsystem could employ a specialized processor especially suited for a particular task

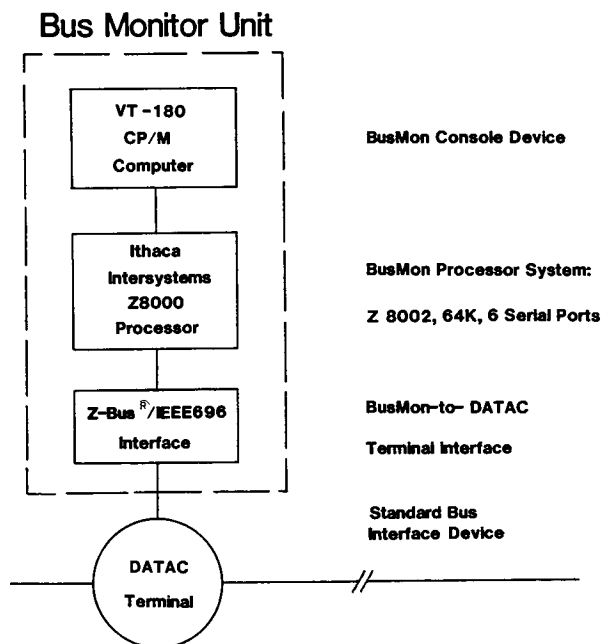
*MINIMUM REPLACEMENT TIMES- line bus configuration means plug-in replacement possible

*IMPROVED FLEXIBILITY- system configuration is independent of the bus itself

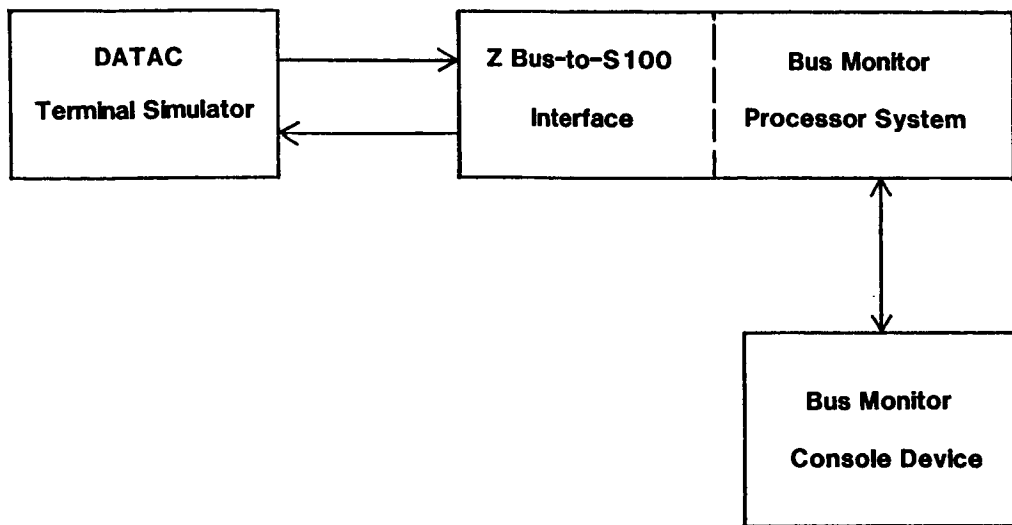
*NO BUS INTERFACE SOFTWARE- handled by bus terminal hardware



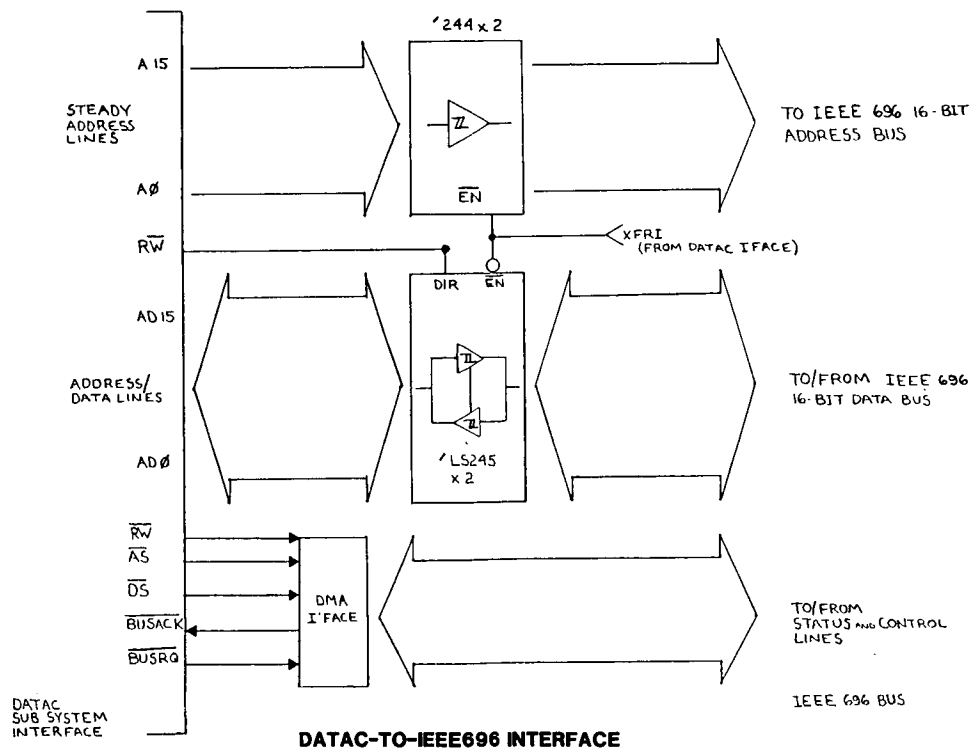
BASIC LAYOUT OF DATAC DATA BUS

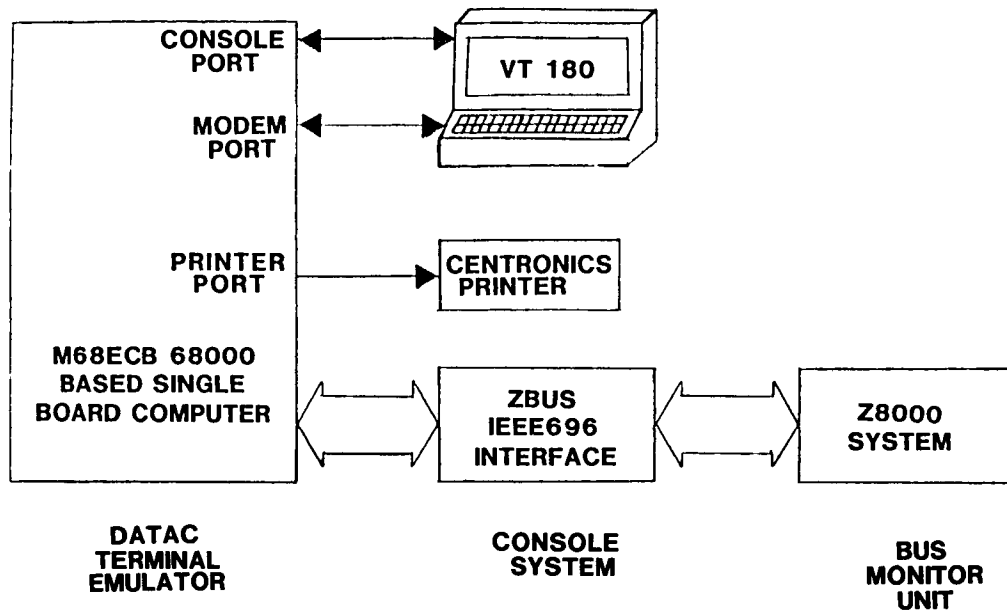


Proposed Bus Monitor Test Configuration



Bus Monitor Software Development System





BUS MONITOR SOFTWARE TEST SYSTEM